



US006185681B1

102e

(12) **United States Patent**
Zizzi

(10) **Patent No.:** **US 6,185,681 B1**
(b5) **Date of Patent:** **Feb. 6, 2001**

(54) **METHOD OF TRANSPARENT ENCRYPTION AND DECRYPTION FOR AN ELECTRONIC DOCUMENT MANAGEMENT SYSTEM**

(76) Inventor: Stephen Zizzi, 2030 Main St., Ste. 1300, Irvine, CA (US) 92612

(*) Notice: Under 35 U.S.C. 154(b), the term of this patent shall be extended for 0 days.

(21) Appl. No.: 09/074,191

(22) Filed: **May 7, 1998**

(51) Int. Cl. **7** H04L 9/00

(52) U.S. Cl. **713/165**

(58) Field of Search **713/173, 159, 713/155, 168, 165, 164**

(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | |
|-----------|---------|--------------------|-----------|
| 4,203,166 | 5/1980 | Ehrsam et al. | 375/2 |
| 4,633,430 | 12/1986 | Cooper | 364/900 |
| 4,956,769 | 9/1990 | Smith | 364/200 |
| 5,163,147 | 11/1992 | Orita | 395/600 |
| 5,243,518 | 9/1993 | Holt et al. | 364/419.1 |
| 5,263,157 | 11/1993 | Janis | 395/600 |

| | | | |
|-----------|----------|-----------------------|------------|
| 5,291,405 | 3/1994 | Kohari | 364/419.19 |
| 5,481,700 | 1/1996 | Thuraisingham | 395/600 |
| 5,530,907 | 6/1996 | Pavey et al. | 395/889 |
| 5,572,673 | 11/1996 | Shurts | 395/186 |
| 5,584,022 | 12/1996 | Kikuchi et al. | 395/609 |
| 5,623,653 | 4/1997 | Matsuno et al. | 395/609 |
| 5,649,185 | 7/1997 | Antognini et al. | 395/609 |
| 5,666,490 | 9/1997 | Gillings et al. | 395/200.68 |
| 5,748,738 | * 5/1998 | Bisbee et al. | 713/176 |
| 5,778,071 | * 7/1998 | Caputo et al. | 713/159 |

* cited by examiner

Primary Examiner—Tod R. Swann

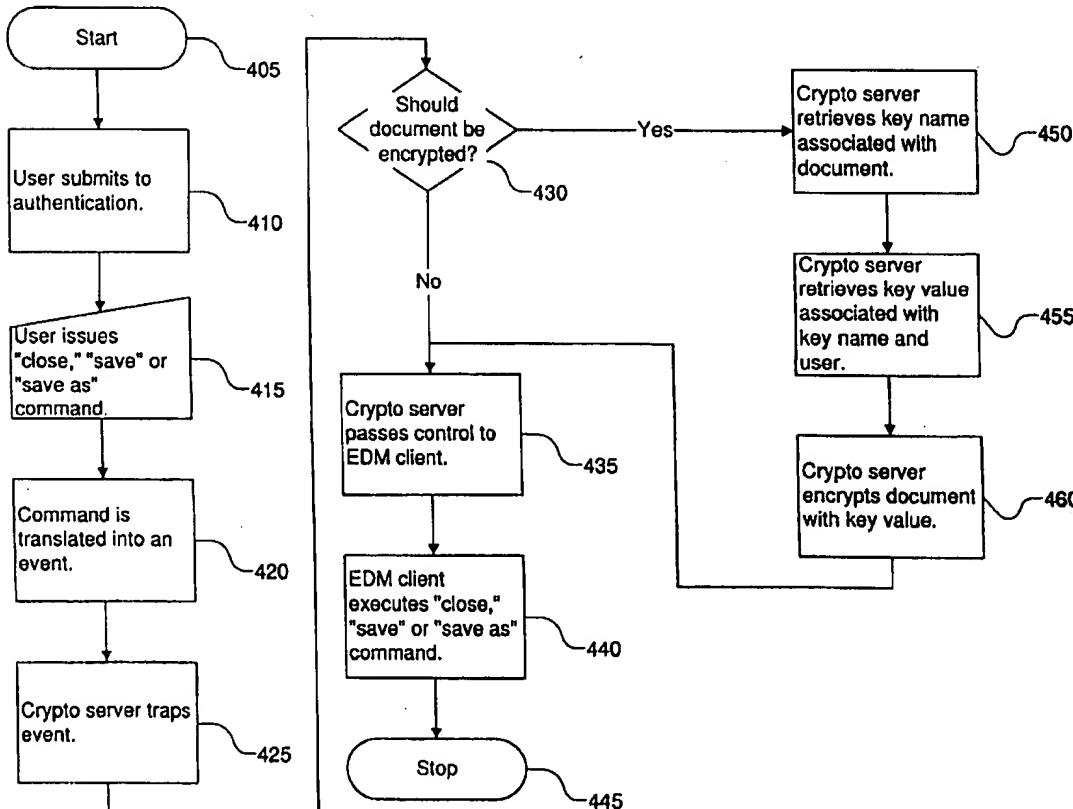
Assistant Examiner—Matthew Smithers

(74) Attorney, Agent, or Firm—Steve C. Sereboff; Roy L. Anderson

(57) **ABSTRACT**

Cryptographic methods and systems are disclosed. The cryptographic methods provide transparent encryption and decryption of documents in an electronic document management system. The cryptographic system adds a software module to an electronic document management system which traps file I/O events and performs cryptographic functions on the relevant documents before passing control back to the electronic document management system.

30 Claims, 5 Drawing Sheets



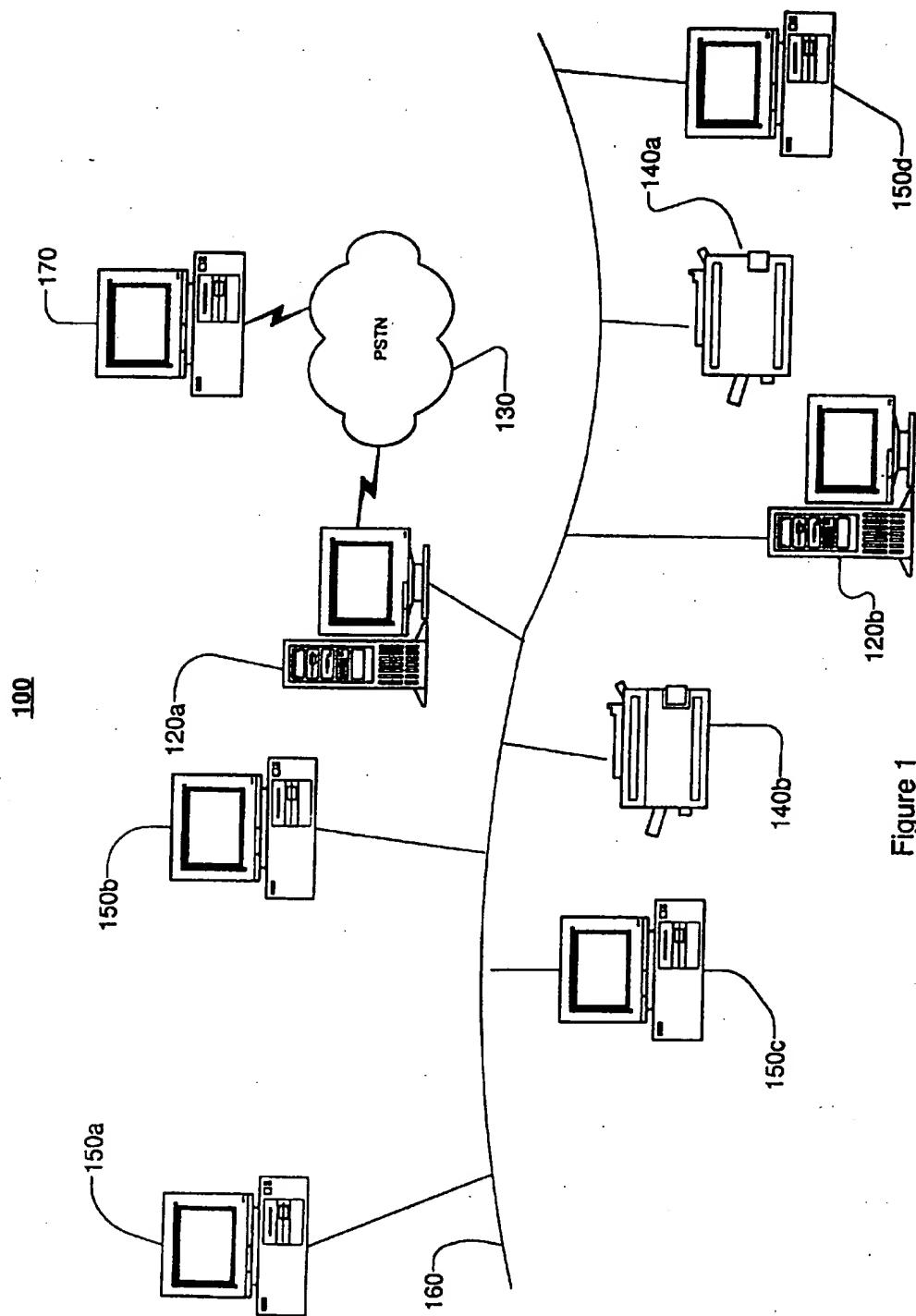


Figure 1

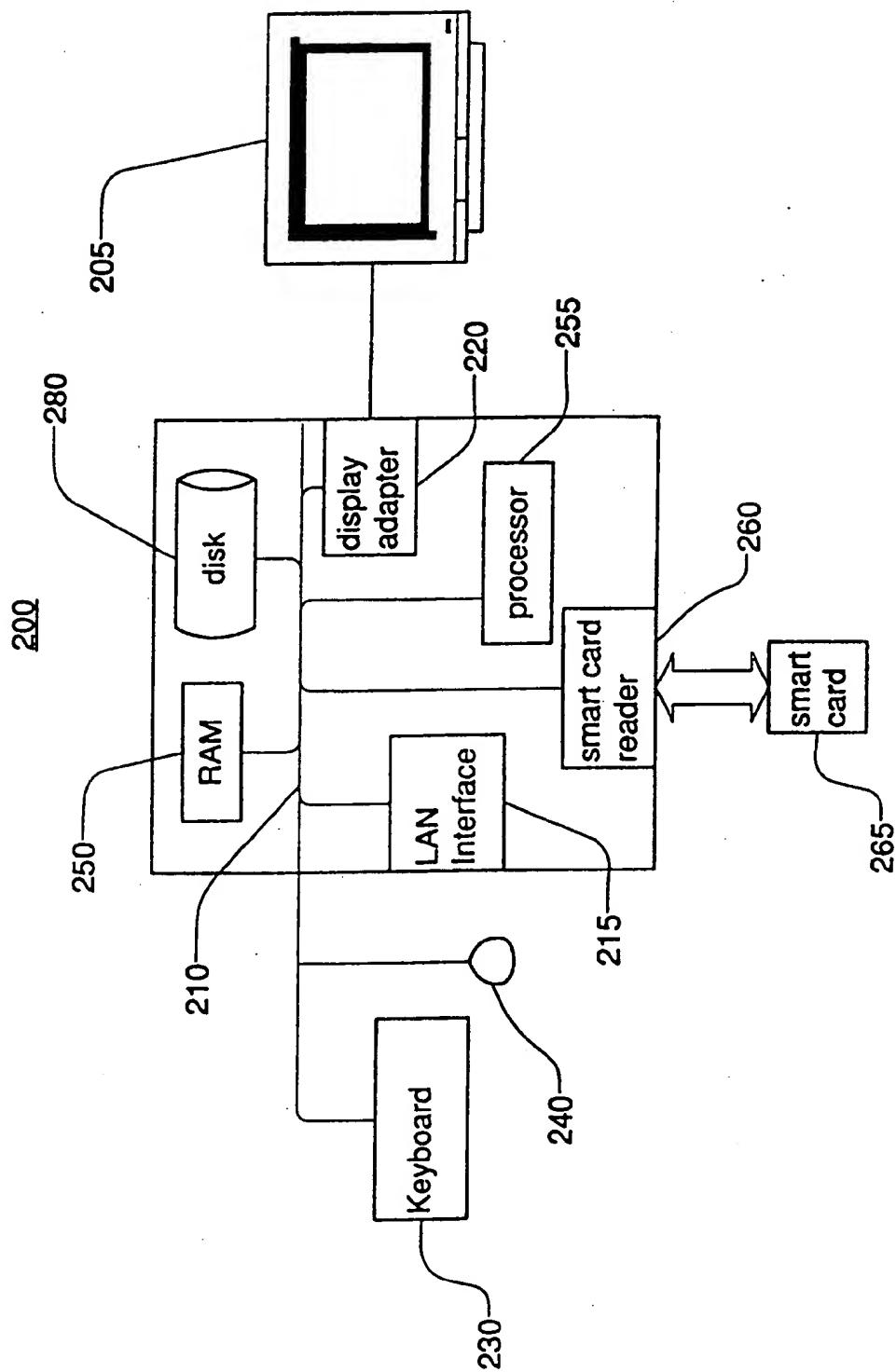


Figure 2

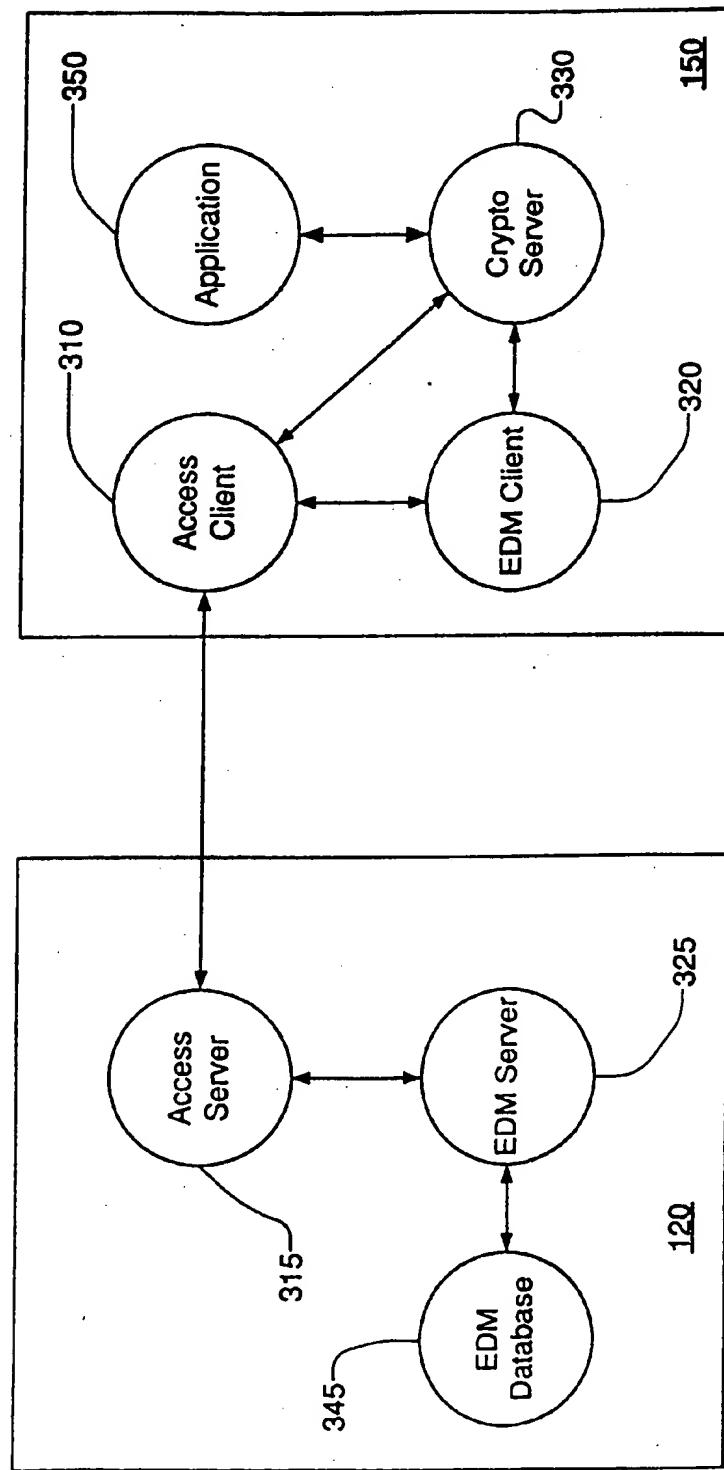


Figure 3

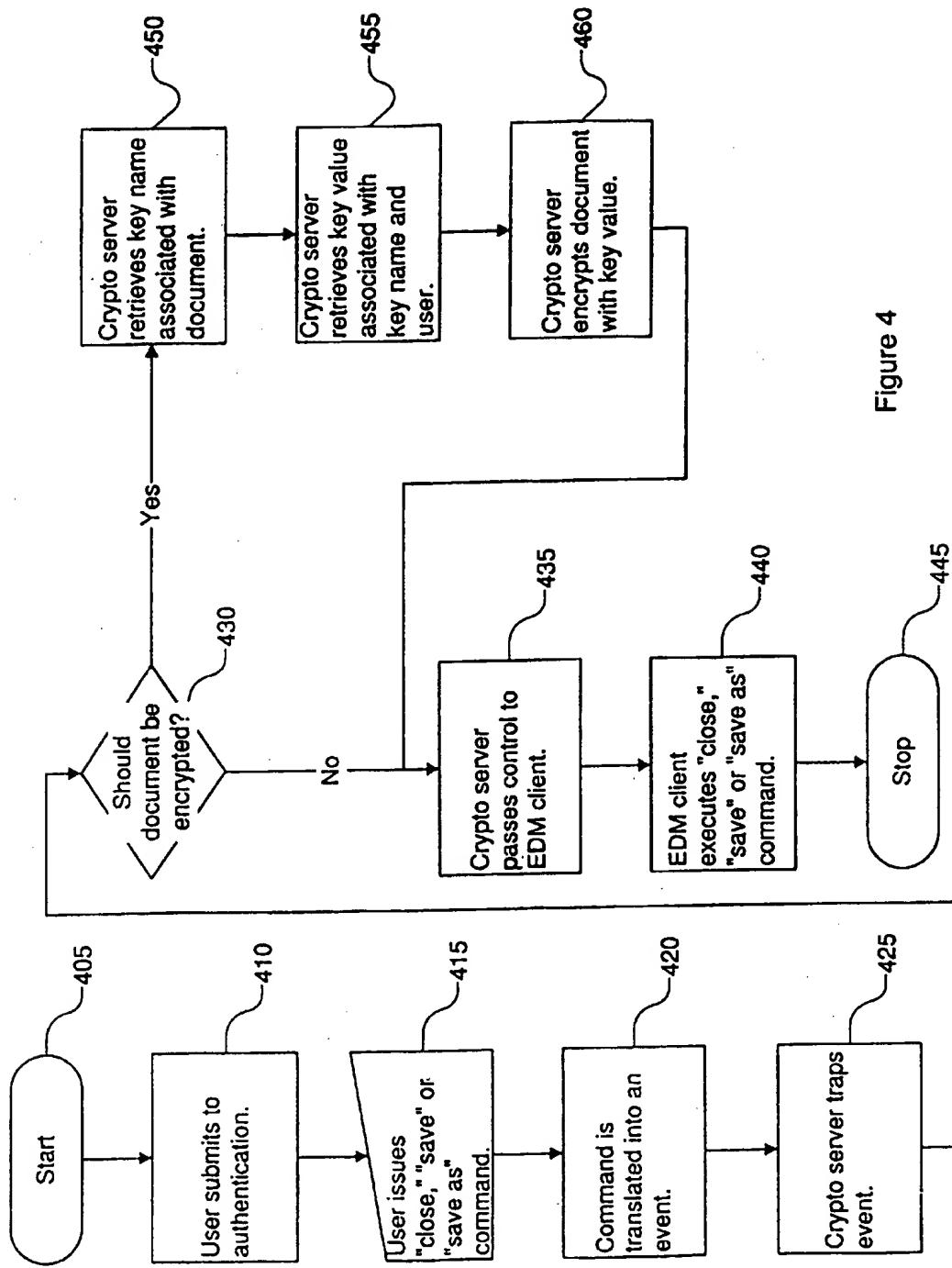


Figure 4

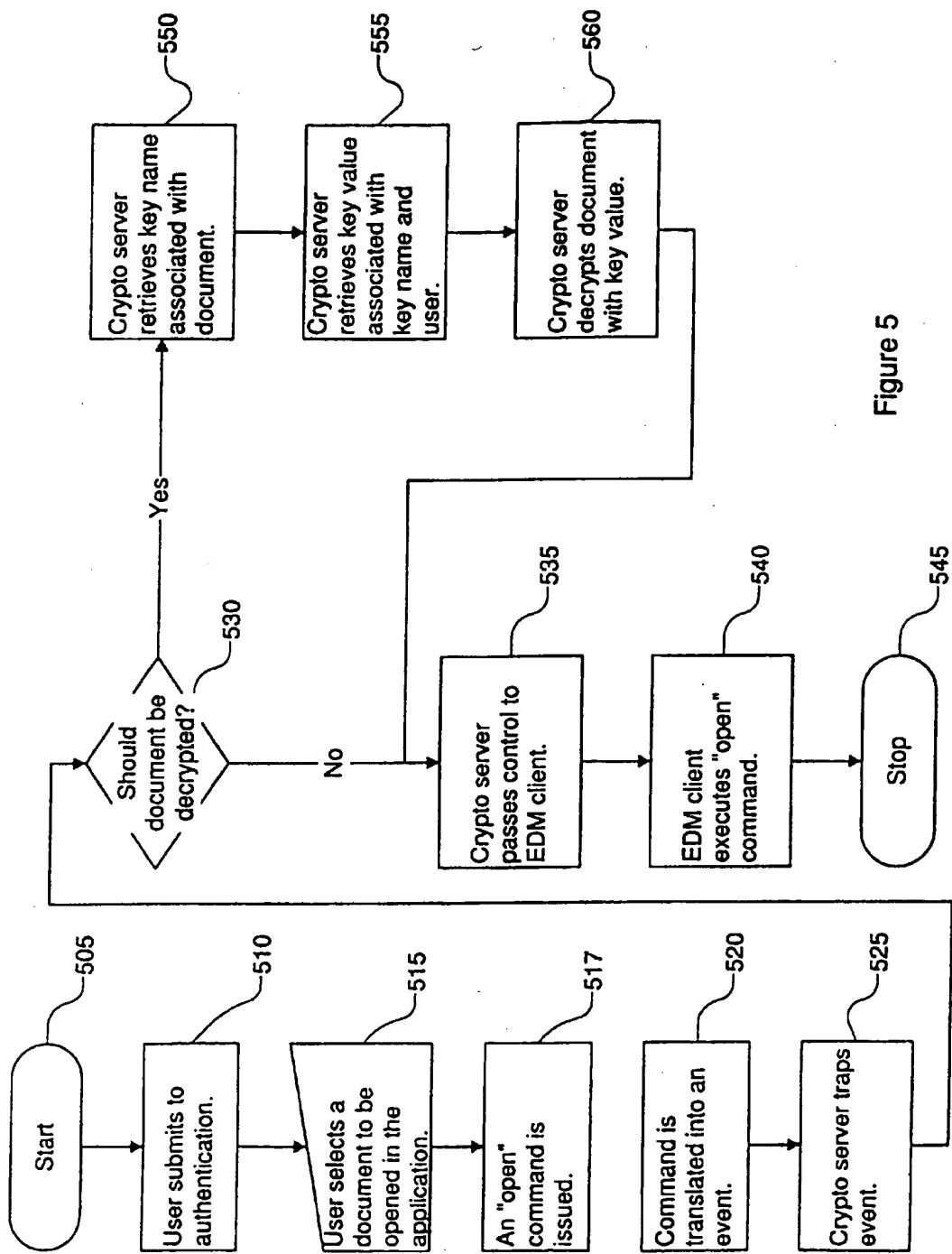


Figure 5

**METHOD OF TRANSPARENT ENCRYPTION
AND DECRYPTION FOR AN ELECTRONIC
DOCUMENT MANAGEMENT SYSTEM**

**NOTICE OF COPYRIGHTS AND TRADE
DRESS**

A portion of the disclosure of this patent document contains material which is subject to copyright protection. This patent document may show and/or describe matter which is or may become trade dress of the owner. The copyright and trade dress owner has no objection to the facsimile reproduction by any one of the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright and trade dress rights whatsoever.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to cryptographic systems and electronic document management systems.

2. Description of Related Art

Global access of electronic information can be critical for even the smallest of businesses today. Very few companies operate solely within the boundaries of a single location or their employee list. Over the last 25 years technology has rapidly advanced and expanded these boundaries. The advent of such technologies as the Internet, intranets, extranets, and e-mail have made the electronic transfer of information common place in businesses today. Management of business information is critical to the success of modern businesses. A technology known as Electronic Document Management (EDM) aims to provide organizations with the ability to find any document, created in any application, by anyone, at any time, dealing with any subject, at any place in the world. EDM includes managing multiple versions of a document. PC DOCS, Inc. (Burlington, Mass.) is one of the world's leading providers of EDM solutions. With the advanced technology of EDM comes a wide variety of information that has varying economic values and privacy aspects. Users may not know what information is monitored or intercepted or who is using their computer.

An electronic document management system (EDMS) is a combination of databases, indexes, and search engines utilized to store and retrieve electronic documents distributed across an organization. An EDMS is designed to provide the structure required for an organization to properly manage and share its electronic document resources.

A wide array of information is typically stored in a company's EDMS. This includes:

- strategic and corporate plans;
- proprietary product and service information;
- confidential legal documents;
- private health information; and
- private employment information.

As companies increase the efficiency of accessing more information, their security risks also increase. According to a recent survey by Ernest & Young LLP:

74% of the respondents said their security risks have increased over the prior two years; more than a quarter said that their security risks have increased at a faster rate than the growth of their computing;

55% of the respondents lacked confidence that their computer systems could withstand an internal attack

71% of security professionals are not confident that their organizations are protected from external attack; and two-thirds of the respondents reported losses resulting from a security breach over the prior two years.

The bottom line is simple—the more information available, the more security needed.

It has been said that "There is no need to break the window of a house if the front door is unlocked." This saying certainly applies to computer security. The "unlocked doors" in electronic information security include:

- e-mail;
- electronic document management (including non-EDMS file systems); and
- stolen hardware.

One of the fastest growing means of communication today is e-mail. It is estimated that over one million e-mail messages pass through the Internet every hour. E-mail provides a quick, economical, easy to use method of sharing both thoughts and electronic information. Unfortunately, e-mail is like an electronic postcard for the world to see. It is transmitted across the Internet using the Simple Mail Transfer Protocol (SMTP). This protocol has virtually no security features. Messages and files can be read by anyone who comes into contact with them.

The number of documents managed by organizations increases daily. Knowledge is becoming the most important product for companies today. As EDM enhances a company's productivity and efficiency to manage that knowledge it also exposes that company to unauthorized access to that knowledge. The typical EDMS solely relies on password protection for security.

The value of the approximately 265,000 portable computers (laptops, notebooks, palmtops) reported stolen in 1996 was \$805 million, a 27% increase from 1995. However, the data on these portable computers is worth much more than the hardware itself. It is critical that the data stored on any type of hardware, whether it is a desktop computer, portable computer or server, must be properly secured from any unauthorized access.

Some of the "locks" used for electronic information security include:

- passwords,
- firewalls,
- smart cards, and
- encryption.

Passwords are often used to prevent unauthorized individuals from accessing electronic data. Passwords may also be used to link activities that have occurred to a particular individual. The problem with passwords is that if any unauthorized party steals or guesses a password, the security of the computer system may be severely compromised. Passwords are wholly inadequate for file archiving.

Systems using firewalls prevent intruders from accessing the firm's internal systems. Password-based firewall systems do not provide positive user identification nor do they protect electronic data that is stored on a server, has left the firm on a portable computer, is sent via e-mail over the Internet, or is stored on a floppy disk.

The typical smart card is a self-contained, tamper resistant, credit card size device that serves as a storage device and is equipped with an integrated microprocessor chip and non-volatile electronic memory. The smart card processes information on the integrated microprocessor chip. Security is enhanced because the user must have the smart card along with the user's confidential information (e.g., a password) to gain access to their computer files.

Passwords are kept off computer hosts and on the smart card to enhance security. Smart cards typically can only be accessed with a user-defined password. Many smart cards include a lock-out feature so that failed attempts at the smart card password will lock the card out to prevent any unauthorized or fraudulent use of the smart card. ISO 7816 compliant smart cards and smart card readers follow industry standards.

Increasingly, information technology professionals are turning to encryption technologies to ensure the privacy of business information. Encryption can provide confidentiality, source authentication, and data integrity. Unfortunately encryption generally is cumbersome and difficult to use. A major obstacle for the implementation of encryption technologies has been their disruption to the users' workflow.

Encryption is a process of scrambling data utilizing a mathematical function called an encryption algorithm, and a key that affects the results of this mathematical function. Data, before becoming encrypted, is said to be "clear text." Encrypted data is said to be "cipher text." With most encryption algorithms, it is nearly impossible to convert cipher text back to clear text without knowledge of the encryption key used. The strength of the encrypted data is generally dependent upon the encryption algorithm and the size of the encryption key.

There are two types of encryption: symmetric (private key) and asymmetric (public key).

Private key encryption uses a common secret key for both encryption and decryption. Private key encryption is best suited to be used in trusted work groups. It is fast and efficient, and properly secures large files. The leading private key encryption is DES (Data Encryption Standard). DES was adopted as a federal standard in 1977. It has been extensively used and is considered to be strong encryption. Other types of private key encryption include: Triple-DES, IDEA, RC4, MD5, Blowfish and Triple Blowfish.

Public key encryption uses a pair of keys, one public and one private. Each user has a personal key pair, and the user's public (or decryption) key is used by others to send encrypted messages to the user, while the private (or decryption) key is employed by the user to decrypt messages received. Public key encryption and key generation algorithms include the public domain Diffie-Hellman algorithm, the RSA algorithm invented by Rivest, Shamir and Adleman at the Massachusetts Institute of Technology (MIT), and the Pretty Good Privacy algorithm (PGP) developed by Phil Zimmermann. Because of their mathematical structure, public key encryption is slower than most private key systems, thus making them less efficient for use in a trusted network or for encrypting large files.

Although these private key and public key encryption algorithms do a good job at maintaining the confidentiality of the encrypted matter, they have numerous problems. The biggest obstacle to adoption of any type of encryption system has been ease of use. Typical encryption systems are very cumbersome. They require a user to interrupt their normal work flow, save their clear text document, activate the separate encryption software, and save the cipher text document under a different name. Where the subject document is ordinary e-mail contents, the process is especially cumbersome, because the clear text must first be created in a separate application, then encrypted, then attached to the e-mail message.

A major concern in computing today is "total cost of ownership," or TCO. TCO recognizes that while a program might be inexpensive (or even free in the case of PGP for

non-commercial use), there are significant costs in using the software. This includes the cost of installation, training, lost productivity during use and from bugs, and maintenance.

Even where one of the typical encryption systems might satisfy a user's TCO needs, they may not even be an available option. For example, typical EDMSes are self-contained and are not compatible with typical encryption systems.

It is therefore the object of the invention to provide a document encryption and decryption system which solves these problems. It is a further object to provide a document encryption and decryption system which works with minimal disruption of a user's normal workflow. It is a further object to provide a document encryption and decryption system which is compatible with EDMSes. It is a further object to provide a document encryption and decryption system which minimizes TCO. It is a further object to provide a document encryption and decryption system which takes advantage of the features of smart cards which are not available from pure on-line security systems.

SUMMARY OF THE INVENTION

The previously described objects are achieved in a method of encrypting documents and a method of decrypting documents in which the cryptographic process is largely transparent to the user.

In the encryption method, after a user issues a "close," "save" or "save as" command for a document, the command is translated into an event, and a crypto module traps the event. The crypto module then obtains an encryption key value and encrypts the document using the encryption key value. With the document encrypted, the crypto module passes control to an electronic document management system which executes the "close," "save" or "save as" command.

In the decryption method, after the user selects a document to be opened, an "open" command issues and is translated into an event. The crypto module traps the event, retrieves a decryption key value and decrypts the document using the decryption key value. The crypto module then passes control to the electronic document management system which executes the "open" command so that the document is opened in the appropriate application program.

According to another aspect of the invention, an electronic document management system distributed between a file server and a workstation is enhanced with transparent cryptography. The file server includes an access server comprising software for handling user authentication and file system access control for the file server. The workstation includes an access client comprising software for enabling a user to sign on to the file server and obtain access to the file system on the file server. There is also an EDM server in the file server comprising software for controlling an EDM database and EDM indexes to the EDM database, and an EDM client in the workstation comprising software for interfacing the workstation to the EDM server and thereby allowing access by a user at the workstation to the EDM database. A crypto server is also included in the electronic document management system of the invention, and the crypto server comprises software for intercepting I/O requests by the application and transparently handling encryption of the documents and decryption of encrypted documents.

Still further objects and advantages attaching to the device and to its use and operation will be apparent to those skilled in the art from the following particular description.

DESCRIPTION OF THE DRAWINGS

Further objects of this invention, together with additional features contributing thereto and advantages accruing therefrom, will be apparent from the following description of a preferred embodiment of the present invention which is shown in the accompanying drawings with like reference numerals indicating corresponding parts throughout and which is to be read in conjunction with the following drawings, wherein:

FIG. 1 is a block diagram of a computer network in accordance with the invention.

FIG. 2 is a block diagram of a general purpose computer in accordance with the invention.

FIG. 3 is a functional block diagram of a cryptographic system in accordance with the invention.

FIG. 4 is a flowchart of an encryption process in accordance with the invention.

FIG. 5 is a flowchart of a decryption process in accordance with the invention.

These and additional embodiments of the invention may now be better understood by turning to the following detailed description wherein an illustrated embodiment is described.

DETAILED DESCRIPTION OF THE INVENTION

Throughout this description, the preferred embodiment and examples shown should be considered as exemplars, rather than limitations on the apparatus and methods of the present invention.

FIG. 1 shows a local area network (LAN) 100. To network communication lines 160 are coupled a number of workstations 150a, 150b, 150c, 150d. A number of file servers 120a, 120b also are coupled to the network communication lines 160. The network communication lines 160 may be wire, fiber, or wireless channels as known in the art. A user at any of the workstations 150 preferably may log on to at least one file server 120 as known in the art, and in some embodiments a workstation 150 may be logged on to multiple file servers 120. One or more remote workstations 170 may be provided for dial-in access to the server 120a through the public switched telephone network 130 or other remote access means. Network printers 140a, 140b are also provided for printing documents. The network 100 may also include hubs, routers and other devices (not shown).

FIG. 2 shows a general purpose computer 200 which is representative of the workstations 150 and file servers 120. The computer 200 preferably includes an Intel Corporation (San Jose, Calif.) processor 255 and runs a Microsoft Corporation (Redmond, Wash.) Windows operating system. In conjunction with the processor 255, the computer 200 has a short term memory 250 (preferably RAM) and a long term memory 280 (preferably a hard disk) as known in the art. The computer 200 further includes a LAN interface 215, a display 205, a display adapter 220, a keyboard 230, a mouse 240, a smart card reader 260 and a bus 210 as known in the art.

The smart card reader 260 preferably complies with ISO 7816, a standard available from the American National Standards Institute (ANSI). To interface the smart card reader 260 to the computer's Windows operating system and other software, the computer 200 preferably includes an API provided by the smart card reader manufacturer. Alternatively, the computer 200 may include Microsoft's smart card API—SCard COM, available at www.microsoft.com/smartscard.

A user's smart card 265 preferably stores a unique user ID and password and a definable hierarchy of encryption keys. The hierarchy preferably forms a table wherein a key name is associated with each key value in the table, and the table may store both encryption keys and decryption keys as necessary for the selected cryptographic algorithms. It should be appreciated that, in private key cryptography, the same key value is used for both encryption and decryption.

Although something as simple as a user ID/password scheme could be used with the keys stored in the disk 280 or memorized by the user, a data reader device and portable data storage device such as the smart card reader 260 and smart card 265 are preferred. Instead of the smart card reader 260 and smart card 265, there could be provided, for example, a biometric recognition system, wireless identification devices, hand held tokens, etc. Preferably, the portable data storage device can securely store one or more encryption and decryption keys. However, a biometric recognition system may provide key selection based on inherent biometric features, eliminating the need to actually store keys in a component external to the computer 200. Where the portable data storage device is used solely as a source of positive identification (i.e., authentication), the keys may be stored on the 120 file server for example and accessed through a certificate mechanism.

Before proceeding, a few terms are defined. By "file server" it is meant a computer which controls access to file and disk resources on a network, and provides security and synchronization on the network through a network operating system. By "server" it is meant hardware or software which provides network services. By "workstation" it is meant a client computer which routes commands either to its local operating system or to a network interface adapter for processing and transmission on the network. By "client" it is meant software which is serviced by a server. A workstation may function as a server by including appropriate software, and may be for example, a print server, archive server or communication server. By "software" it is meant one or more computer interpretable programs and/or modules related and preferably integrated for performing a desired function. By "document" it is meant a named, structural unit of text, graphics and/or other data that can be stored, retrieved and exchanged among systems and users as a separate unit.

Referring now to FIG. 3, there is shown a conceptual block diagram of several functional units relevant to the invention which operate within the file server 120 and workstation 150. The workstation 150 includes at least one application 350. The application 350 is a collection of software components used to perform specific types of user-oriented work and may be, for example, a graphic editor, a word processor or a spreadsheet.

As is typical in the art, the workstation 150 obtains access to the file server 120 through a user ID and password system 55 which extends to the file system on the file server 120. The file server has an access server 315 for handling the filer server's user authentication and access control duties, and the workstation 150 includes an access client 310 through which a user signs on to the file server 120. In the preferred embodiment, the access server 315 is a part of Windows NT Server, and the access client 310 is a part of Windows 95 and Windows NT Workstation. Other operating systems such as Unix and Novell Netware also include access servers and access clients for providing user authentication and file level security.

Within the file server 120 there is preferably an EDM server 325. To interface with the EDM server 325, the

workstation 150 includes an EDM client 320, sometimes referred to as an "EDM plug-in." The EDM server 325 controls an EDM database 345 and EDM indexes (not shown), and preferably provides EDM search engines. The EDM database 345 itself may be distributed, for example across file systems and file servers, and may be entirely or partially in the workstation 150. The EDM server 325 may include a database server such as a SQL server for interfacing to the EDM database 345. The EDM client 320 provides the workstation with an interface to the EDM server and therefore allows access by a user at the workstation 150 to the EDM database 345, indexing and search services provided by the EDM server 325.

The EDMS of the preferred embodiment is SQL-based. Thus, the EDM database 345 comprises a SQL database, the EDM server 325 comprises a SQL server, and the EDM client 320 comprises a SQL plug-in. The SQL database stores file and file location information. A "repository," which could be considered part of the EDM database 345, stores the files, and is managed and distributed using techniques known in the art. In older EDM systems, the SQL plug-in comprises special software which adapted particular popular applications for use with the EDMS. However, with the promulgation of the Open Document Management Architecture (ODMA) specification, applications are available which operate seamlessly with many contemporary EDM systems. Under ODMA, the EDM plug-in registers itself so that it handles file I/O.

The EDM server 325, EDM database 345 and EDM client 320 are described herein as wholly separate from the respective operating systems of the file server 120 and workstation 150. However, much if not all of the EDM server 325, EDM database 345 and EDM client 320 could be fully integrated into and even become a part of the respective operating systems. In such an embodiment, the EDMS is just another part of an operating system's general file and data management features.

As can be seen, the access server 315 and the access client 310 functionally reside between the EDM server 325 and the EDM client 320, thereby separating the EDM server 325 and EDM client 320 with a measure of security. This aspect of FIG. 3 is the typical prior art configuration, and it provides file-level security for documents in the EDM database 345 controlled by the EDM server 325.

Positioned functionally between the application 350 and the EDM client 310 is a crypto server 330. In typical prior art systems, the application 350 would communicate directly with the EDM client 310. However, in accordance with the invention, the crypto server 330 is functionally disposed between the application 350 and the EDM client 310, and intercepts or traps I/O requests by the application which otherwise would be intercepted or trapped by the EDM client 310.

The crypto server 330 of the invention is a software module which transparently handles the encryption of documents and the decryption of encrypted documents, making encryption and decryption simple and easy to use. The crypto server 330 handles encryption and decryption without requiring user input and without normally displaying status information during normal encryption and decryption operations. Preferably, the user or a system administrator may establish a system-level configuration determinative of when error messages should be displayed. Preferably, also, the system administrator may create and maintain a file administration table in the EDM database 345 which defines criteria for which files are to be encrypted and which key to

use. The crypto server 330 utilizes the file administration table, for example, to determine if a new file should be encrypted, and which encryption key to use to encrypt the new file. The crypto server 330 preferably utilizes and updates an encrypted files table in the EDM database 345 which lists each encrypted file.

The crypto server 330 may itself comprise a number of functional units. For example, the crypto server 330 preferably includes interfaces to one or more cryptographic systems, such as those described in the Description of the Related Art section above. The crypto server 330 preferably also includes an interface to the smart card reader 260 (FIG. 2) for reading the smart card 265. The smart card 265 preferably is used to keep the encryption and decryption keys separate from the workstation 150 and provide positive user identification. The crypto server 330 also works with the access client 310 in performing user authentication and access. In particular, the typical prior art user access process is enhanced by requiring that the user enter a user ID and password which are stored on the user's smart card 265.

Turning now to FIG. 4, there is shown a flowchart of the encryption process in accordance with the invention. After the process begins (step 405), it is preferred that the user submit to authentication by the access client 310 and access server 315 (step 410). The authentication step is preferably performed when the user signs onto the workstation 150. Preferably, the user must insert his smart card 265 into the smart card reader 260 and enter the user ID and password stored on the smart card 265. Once authenticated, the smart card 265 then makes available, as needed, the encryption and decryption key information stored therein.

At some point after the user has been authenticated, the user will be working on a document in the application 350, and at some point issue a "close," "save" or "save as" command as known in the art (step 415). The command is then translated into an "event" (step 420), and the crypto server 330 traps this event (step 425). Techniques for translating commands into events and trapping events are well known in the art and are typically different for each operating system. In Windows, the event translation step comprises generating an event message.

The trapped event has the effect of alerting the crypto server 330 that it may be necessary to encrypt the document. However, preferably before encrypting the document, the crypto server 330 tests whether the document should be encrypted (step 430). Preferably, at least three different tests are performed.

In the first test, the crypto server 330 tests whether the user has been authenticated. The first test is relatively simple. Where the smart card 265 or similar means is used for storing keys, this test is necessary because the keys will not even be available unless the user was authenticated.

In the second test, the crypto server 330 tests whether the document was already encrypted when it was opened by the application 350. By default, a document which was already encrypted when opened should be encrypted when closed or saved.

In the third test, the crypto server 330 tests whether the EDM database 345 has an indicator that the document should be encrypted. As described above, the EDM database 345 includes a list of encrypted documents in an encrypted files table. The EDM database 345 preferably also includes criteria for new documents which indicate whether new documents, when the criteria are met, should be encrypted. The criteria are preferably stored in the file administration table described above. To perform the third test, the crypto

server 330 passes a database query to the EDM client 320 to have the EDM server 325 query the EDM database 345. For existing files, the query is directed to the encrypted files table. For new files, the query is directed to the file administration table. The EDM server 325 then passes the results of the test back to the EDM client 320, which provides the test results to the crypto server 330.

If for any reason the document is not to be encrypted, then the crypto server 330 passes control to the EDM client 320 which performs the "close," "save" or "save as" command on the unencrypted document. Alternatively, the decision not to encrypt, for one or more reasons, may result in an error message being displayed to the user, and may result in the document not being closed or saved. At this point, for documents which are not to be encrypted, the method is complete (step 445).

If, in step 430, the document is to be encrypted, then the crypto server 330 preferably obtains an encryption key name which is associated with the document (step 450).

The crypto server 330 then uses the encryption key name to retrieve an encryption key value which is associated with the encryption key name (step 455). For most encryption algorithms, the encryption key is a multi-digit number which is difficult to remember and even difficult to transcribe. The encryption key name is preferably an alphanumeric descriptor which may be used by the user and/or system administrator for administering the encryption key value. Preferably, the encryption key value is also related to the identify of the user, and this is accomplished by retrieving the encryption key value from the key table stored in the smart card 265 which is associated with the relevant encryption key name.

Once the crypto server 330 has the encryption key value, the crypto server 330 then encrypts the document with the encryption key value (step 460), and passes control to the EDM client (step 435) so that the document may be saved (step 440). At this point, for documents which are to be encrypted, the method is complete (step 445).

Turning now to FIG. 5, there is shown a flowchart of the decryption process in accordance with the invention. After the process begins (step 505), it is preferred that the user submit to authentication (step 510). Authentication (step 505) preferably is the same for encryption and decryption.

At some point after the user has been authenticated, the user will wish to open a document into the application 350 (step 515). The file open command may be issued from within the application 350 or may be issued by a second application, with the nature of the document such that the application 350 will actually open the document and provide access to the document's contents. In any case, once the user selects a document to be opened, an "open" command is issued (step 517). The open command is then translated into an event (step 520), and the crypto server 330 traps this event (step 525).

The trapped event has the effect of alerting the crypto server 330 that it may be necessary to decrypt the document. However, preferably before decrypting the document, the crypto server 330 tests whether the document should be decrypted (step 430). Preferably, these tests are complementary to those described above with respect to the encryption process.

If for any reason the document is not to be decrypted, then the crypto server 330 passes control to the EDM client 320 which performs the "open" command. Alternatively, the decision not to decrypt, for one or more reasons, may result in an error message being displayed to the user, and may result in the document not being opened. At this point, for

documents which are not to be decrypted, the method is complete (step 545).

If, in step 530, the document is to be decrypted, then the crypto server 330 preferably obtains a decryption key name which is associated with the document (step 550). The decryption key name is preferably obtained from the file's header or from the encrypted files table.

The crypto server 330 then uses the decryption key name to retrieve a decryption key value which is associated with the decryption key name (step 555). Preferably, the decryption key value, like the encryption key value, is also related to the identify of the user, and this is accomplished by retrieving the decryption key value from the key table stored in the smart card 265 and associated with the decryption key name.

Once the crypto server 330 has the decryption key value, the crypto server 330 then decrypts the document with the decryption key value (step 560), and passes control to the EDM client (step 535) so that the decrypted copy of the document may be opened into the application (step 540). At this point, for documents which are to be decrypted, the method is complete (step 545).

Although exemplary embodiments of the present invention have been shown and described, it will be apparent to those having ordinary skill in the art that a number of changes, modifications, or alterations to the invention as described herein may be made, none of which depart from the spirit of the present invention. All such changes, modifications and alterations should therefore be seen as within the scope of the present invention.

It is claimed:

1. A method of encrypting an electronic document which is open in an application program running in a general purpose computer, the general purpose computer including a display, a user input device, a crypto module and a processor, the method comprising:

- (a) from within the application program running in the general purpose computer, a user issuing one of a "close," "save" or "save as" command for the document using the user input device;
- (b) automatically translating the command into an event;
- (c) the crypto module automatically trapping the event;
- (d) the crypto module automatically obtaining an encryption key value;
- (e) the crypto module automatically encrypting the document using the encryption key value;
- (f) the crypto module automatically passing control to an electronic document management system; and
- (g) the electronic document management system executing the issued "close," "save" or "save as" command; whereby the electronic document is automatically encrypted.

2. A method of encrypting a document as set forth in claim 1 wherein the electronic document management system comprises a SQL database, a SQL database server and a SQL database client, the SQL database client being disposed in the general purpose computer.

3. A method of encrypting a document as set forth in claim 1 where step (d) comprises the steps of the crypto module determining if the document should be encrypted, and if not, then skipping step (e), and if so, then:

- the crypto module retrieving an encryption key name associated with the document; and
- the crypto module retrieving the encryption key value associated with the encryption key name.

4. A method of encrypting a document as set forth in claim 3, wherein there are plural encryption key values and at least one encryption key value is associated with the user, the method further comprising the steps of:

the user submitting to an access module for user authentication;
 if the access module does not authenticate the user, then always skipping steps (d) and (e);
 else in step (d), the crypto module retrieving the encryption key value associated with the encryption key name and the user.

5. A method of encrypting a document as set forth in claim 4, the general purpose computer further comprising a data reader device for reading user identification and encryption key values from a portable data storage device, the method further comprising the user presenting the portable data storage device to the data reader device, wherein the access module utilizes information stored in the portable data storage device to authenticate the user, and the encryption key value associated with the user is stored in the portable data storage device.

6. A method of encrypting a document as set forth in claim 5, wherein the data reader device comprises a smart card reader and the portable data storage device comprises a smart card.

7. A method of encrypting a document as set forth in claim 5, wherein the data reader device comprises a biometric recognition system and the portable data storage device comprises the user, wherein the access module utilizes unique information about the user for authentication, and the encryption key value is derived from at least one characteristic of the user.

8. A method of encrypting a document as set forth in claim 1 wherein the electronic document management system comprises a database, the database including an indicator of whether the document should be encrypted, and step (c) further comprises, if the indicator in the database does not indicate that the document is to be encrypted, then skipping steps (d) and (e).

9. A method of encrypting a document as set forth in claim 8, wherein if the indicator in the database does not indicate that the document is to be encrypted, then also skipping steps (f) and (g).

10. A method of encrypting a document as set forth in claim 1 wherein the general purpose computer comprises a workstation, and there is further provided a file server, wherein the crypto module comprises a crypto server on the workstation, the access module comprises an access server on the file server and an access client on the workstation, and the electronic document management system comprises an EDM database on the file server, an EDM server on the file server, and an EDM client on the workstation.

11. A method of encrypting a document as set forth in claim 1 wherein the operating system includes at least part of the electronic document management system.

12. A method of decrypting a document which is to be opened in an application program running in a general purpose computer, the general purpose computer including a display, user input device, a crypto module and a processor, the method comprising:

(a) user selecting the document to be opened in the application program using the user input device;
 (b) an "open" command issuing for the document to be opened in the application program;
 (c) automatically translating the command into an event;
 (d) the crypto module automatically trapping the event;

(e) the crypto module automatically retrieving a decryption key value;

(f) the crypto module automatically decrypting the document using the decryption key value;

(g) the crypto module automatically passing control to an electronic document management system; and

(h) the electronic document management system automatically executing the issued "open" command so that the document is opened in the application program; whereby the document is automatically decrypted.

13. A method of decrypting a document as set forth in claim 12 wherein the electronic document management system comprises a SQL database, a SQL database server and a SQL database client, the SQL database client being disposed in the general purpose computer.

14. A method of decrypting a document as set forth in claim 12 wherein step (e) comprises the crypto module determining if the document should be decrypted, and if not, then skipping step (f), and if so, then:

the crypto module retrieving a decryption key name associated with the document; and
 the crypto module retrieving the decryption key value associated with the decryption key name.

15. A method of decrypting a document as set forth in claim 14, wherein there are plural decryption key values and at least one decryption key value is associated with the user, the method further comprising the steps of:

the user submitting to an access module for user authentication;
 if the access module does not authenticate the user, then always skipping steps (e) and (f);
 else in step (e), the crypto module retrieving the decryption key value associated with the decryption key name and the user.

16. A method of decrypting a document as set forth in claim 15, the general purpose computer further comprising a data reader device for reading user identification and decryption key values from a portable data storage device, the method further comprising the user presenting the portable data storage device to the data reader device, wherein the access module utilizes information stored in the portable data storage device to authenticate the user, and the decryption key value associated with the user is stored in the portable data storage device.

17. A method of decrypting a document as set forth in claim 16, wherein the data reader device comprises a smart card reader and the portable data storage device comprises a smart card.

18. A method of decrypting a document as set forth in claim 16, wherein the data reader device comprises a biometric recognition system and the portable data storage device comprises the user, wherein the access module utilizes unique information about the user for authentication, and the decryption key value is derived from at least one characteristic of the user.

19. A method of decrypting a document as set forth in claim 12 wherein the electronic document management system comprises a database, the database including an indicator of whether the document should be decrypted, and step (d) further comprises, if the indicator in the database does not indicate that the document is to be decrypted, then skipping steps (e) and (f).

20. A method of decrypting a document as set forth in claim 19, wherein if the indicator in the database does not indicate that the document is to be decrypted, then also skipping steps (g) and (h).

21. A method of decrypting a document as set forth in claim 12 wherein the operating system includes at least a part of the electronic document management system.

22. A method of decrypting a document as set forth in claim 12 wherein the general purpose computer comprises a workstation, and there is further provided a file server, wherein the crypto module comprises a crypto server on the workstation, the access module comprises an access server on the file server and an access client on the workstation, and the electronic document management system comprises an EDM database on the file server, an EDM server on the file server, and an EDM client on the workstation.

23. An electronic document management system for storing documents from an application in a workstation and retrieving documents from a file server to the application, the file server having a file system, the electronic document management system comprising:

- (a) an access server in the file server comprising software for handling user authentication and file system access control for the file server;
- (b) an access client in the workstation comprising software for enabling a user to sign on to the file server and obtain access to the file system on the file server;
- (c) an EDM server in the file server comprising software for controlling an EDM database and EDM indexes to the EDM database;
- (d) an EDM client in the workstation comprising software for interfacing the workstation to the EDM server and thereby allowing access by a user at the workstation to the EDM database; and
- (e) a crypto server comprising software for automatically intercepting I/O requests by the application and transparently handling encryption of the documents and decryption of encrypted documents;

wherein the access server and access client are functionally positioned between the EDM server and EDM client, and

the crypto server is functionally positioned between the application and the EDM client.

24. An electronic document management system as set forth in claim 23, wherein the crypto server software includes display commands, the display commands only for displaying error messages to the user.

25. An electronic document management system as set forth in claim 23, wherein the crypto server includes interfaces to plural cryptographic systems.

26. An electronic document management system as set forth in claim 25, the cryptographic systems comprising at least one of RSA, DES, Triple-DES, Blowfish, Triple Blowfish and IDEA.

27. An electronic document management system as set forth in claim 23, the workstation further comprising a data reader device for reading user identification and key values from a portable data storage device, wherein the access client utilizes information stored in the portable data storage device to authenticate the user, and the crypto server obtains key values for encrypting and decrypting the documents from the portable data storage device via the data reader device.

28. An electronic document management system as set forth in claim 27, wherein the data reader device comprises a smart card reader and the portable data storage device comprises a smart card.

29. A program product for use in a general purpose computer that executes program steps recorded in a computer-readable media to perform a method of encrypting an electronic document which is open in an application program running in a general purpose computer, the general purpose computer including a display, a user input device and a processor, the program product comprising:

- a recordable media; and
- a program of computer-readable instructions executable by the computer system to perform method steps comprising:
 - (a) in response to a user issuing one of a "close," "save" or "save as" command for the document using the user input device from within the application program and the command being translated into an event, automatically trapping the event;
 - (b) automatically obtaining an encryption key value;
 - (c) automatically encrypting the document using the encryption key value;
 - (d) automatically passing control to an electronic document management system;

whereby the electronic document management system can then execute the issued "close," "save" or "save as" command and the electronic document is automatically encrypted.

30. A program product for use in a general purpose computer that executes program steps recorded in a computer-readable media to perform a method of decrypting a document which is to be opened in an application program running in the general purpose computer, the general purpose computer including a display, user input device and a processor, the program product comprising:

- a recordable media; and
- a program of computer-readable instructions executable by the computer system to perform method steps comprising:
 - (a) in response to a user selecting the document to be opened in the application program using the user input device and an "open" command issuing for the document to be opened in the application program and the command being translated into an event, automatically trapping the event;
 - (b) automatically retrieving a decryption key value;
 - (c) automatically decrypting the document using the decryption key value; and
 - (d) automatically passing control to an electronic document management system;

whereby the electronic document management system can then execute the issued open command so that the document is opened in the application program and the document is automatically decrypted.

* * * * *